UNITED STATES PATENT APPLICATION

FOR

METHOD AND APPARATUS FOR AGGREGATING INPUT DATA STREAMS

INVENTORS:

Yuen Fai Wong, a citizen of the People's Republic of China
Yu-Mei Lin, a citizen of the United States of America
Richard A. Grenier, a citizen of the United States of America

ASSIGNED TO:

Foundry Networks, Inc., a Delaware Corporation

PREPARED BY:

THELEN REID & PRIEST LLP
P.O. BOX 640640
SAN JOSE, CA 95164-0640
TELEPHONE: (408) 292-5800
FAX: (408) 287-8040

Attorney Docket Number: FOUND-0091 (034103-000036)

Client Docket Number: 91

SPECIFICATION

TITLE OF INVENTION

METHOD AND APPARATUS FOR AGGREGATING INPUT DATA STREAMS

FIELD OF THE INVENTION

[0001] The present invention relates to network interface devices. More particularly, the present invention relates to a method and apparatus for aggregating input data streams from first processors into one data stream for a second processor.

BACKGROUND OF THE INVENTION

[0002] Switched Ethernet technology has continued evolving beyond the initial 10Mbps (bit per second). Gigabit Ethernet technology complying the Institute of Electrical and Electronics Engineers (IEEE) 1000BASE-T Standard (IEEE 802.3 2002-2002) meets demands for greater speed and bandwidth of increasing network traffic. Gigabit over Copper technologies provides high performance in the Enterprise local area network (LAN) and accelerates the adoption of Gigabit Ethernet in various areas, such as server farms, cluster computing, distributed computing, bandwidth-intensive applications, and the like. Gigabit over Copper technologies can be integrated into the motherboard of a computer system, and many server makers are offering integrated Gigabit over Copper ports, which is also referred to as LAN on Motherboard.

[0003] Gigabit Ethernet works seamlessly with existing Ethernet and Fast Ethernet networks, as well as Ethernet adapters and switches. The 1Gbps (i.e., 1000Mbps) speeds of Gigabit Ethernet are 10 times faster than Fast Ethernet (IEEE 100BASE-T), and 100 times faster than standard Ethernet (IEEE 10BASE-T). 10Gigabit Ethernet (10GbE) enables Gigabit to be migrated into an Enterprise LAN by providing the appropriate backbone connectivity. For example, 10GbE delivers a bandwidth required to support access to Gigabit over Copper attached server farms.

[0004] Switch fabrics and packet processors in high-performance broadband switches, such as Gigabit Ethernet switches or line cards, typically run at a fraction of their rated or maximum capacity. That is, typical processing loads do not require the full capacity of the switch fabrics and packet processors. Thus, it would be desirable to provided a scheme to allow such switch fabrics or packet processors to "oversubscribe" data to achieve more efficient usage of the processing capacity, where oversubscription means that the capacity of the data feed is larger than the capacity of data processing or switching.

BRIEF DESCRIPTION OF THE INVENTION

[0005] A method and apparatus aggregate a plurality of input data streams from first processors into one data stream for a second processor, the circuit and the first and second processors being provided on an electronic circuit substrate. The aggregation circuit includes (a) a plurality of ingress data ports, each ingress data port adapted to receive an input data stream from a corresponding first processor, each input data stream formed of ingress data packets, each ingress data packet including priority factors coded therein, (b) an aggregation module coupled to the ingress data ports, adapted to analyze and combine the plurality of input data steams into one aggregated data stream in response to the priority factors, (c) a memory coupled to the aggregation module, adapted to store analyzed data packets, and (d) an output data port coupled to the aggregation module, adapted to output the aggregated data stream to the second processor.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] The accompanying drawings, which are incorporated into and constitute a part of this specification, illustrate one or more embodiments of the present invention and, together with the detailed description, serve to explain the principles and implementations of the invention.

In the drawings:

FIG. 1 is a block diagram schematically illustrating a circuit for aggregating a plurality of input data streams from first processors into one data stream for a second processor in accordance with one embodiment of the present invention.

FIG. 2 is a block diagram schematically illustrating an example of implementation of the aggregation module of the circuit in accordance with one embodiment of the present invention.

FIG. 3 is a block diagram schematically illustrating a circuit for aggregating an input data stream from a first processor into an aggregated data stream for a second processor in accordance with one embodiment of the present invention.

FIG. 4 is a block diagram schematically illustrating a circuit for aggregating a plurality of input data streams from first processors into one data stream for a second processor in accordance with one embodiment of the present invention.

FIG. 5 is a system block diagram schematically illustrating an example in which two data streams from the switching processors are aggregated into one data stream for a packet processing processor by an aggregation circuit in accordance with one embodiment of the present invention.

FIG. 6 is a process flow diagram schematically illustrating a method for aggregating a plurality of input data streams from first processors into one data stream for a second processor in accordance with one embodiment of the present invention.

FIG. 7 is a data flow diagram schematically illustrating the method of aggregating a plurality of data streams along the receive (Rx) data path in accordance with one embodiment of the present invention.

FIG. 8 is a data flow diagram schematically illustrating the method of aggregating a plurality of data streams along the transmit (Tx) data path in accordance with one embodiment of the present invention.

FIG. 9 is a process flow diagram schematically illustrating a method for aggregating a plurality of input data streams from first processors into one data stream for a second processor, in accordance with one embodiment of the present invention.

DETAILED DESCRIPTION

[0007] Embodiments of the present invention are described herein in the context of a method and apparatus for aggregating input data streams. Those of ordinary skill in the art will realize that the following detailed description of the present invention is illustrative only and is not intended to be in any way limiting. Other embodiments of the present invention will readily suggest themselves to such skilled persons having the benefit of this disclosure. Reference will now be made in detail to implementations of the present invention as illustrated in the accompanying drawings. The same reference indicators will be used throughout the drawings and the following detailed description to refer to the same or like parts.

[0008] In the interest of clarity, not all of the routine features of the implementations described herein are shown and described. It will, of course, be appreciated that in the development of any such actual implementation, numerous implementation-specific decisions must be made in order to achieve the developer's specific goals, such as compliance with application- and business-related constraints, and that these specific goals will vary from one implementation to another and from one developer to another. Moreover, it will be appreciated that such a development effort might be complex and time-consuming, but would nevertheless be a routine undertaking of engineering for those of ordinary skill in the art having the benefit of this disclosure.

[0009] In accordance with one embodiment of the present invention, the components, process steps, and/or data structures may be implemented using various types of operating systems (OS), computing platforms, firmware, computer programs, computer languages, and/or general-purpose machines. The method can be implemented as a programmed process running on processing circuitry. The processing circuitry can take the form of numerous combinations of processors and operating systems, or a stand-alone device. The process can be implemented as instructions executed by such hardware, hardware alone, or any combination thereof. The software may be stored on a program storage device readable by a machine.

[0010] In addition, those of ordinary skill in the art will recognize that devices of a less general purpose nature, such as hardwired devices, field programmable logic devices (FPLDs), including field programmable gate arrays (FPGAs) and complex programmable logic devices (CPLDs), application specific integrated circuits (ASICs), or the like, may also be used without departing from the scope and spirit of the inventive concepts disclosed herein.

[0011] In the context of the present invention, the term "network" includes local area networks (LANs), wide area networks (WANs), the Internet, cable television systems, telephone systems, wireless telecommunications systems, fiber optic networks, ATM networks, frame relay networks, satellite communications systems, and the like. Such networks are well known in the art and consequently are not further described here.

[0012] FIG. 1 schematically illustrates a circuit **10** for aggregating a plurality of input data streams from first processors **12** (**12a**, **12b**) into one data stream for a second processor **14** in accordance with one embodiment of the present invention. The circuit **10**, the first processors **12**, and the second processor **14** are provided on an electronic circuit substrate. For example, such an electronic circuit substrate may be a circuit board for a line card, network interface device, and the like.

[0013] As shown in FIG. 1, the circuit **10** includes a plurality of ingress data ports **16** (**16a**, **16b**), an aggregation module **18** coupled to the plurality of ingress data ports **16**, a memory **20** coupled to the aggregation module **18**, and an output data port **22** coupled to the aggregation module **18**. The aggregation module **18** may be implemented by a field programmable logic device (FPLD), field programmable gate array (FPGA), or the like. Each of the ingress data port **16** (**16a** or **16b**) receives an input data stream **24** (**24a** or **24b**) from a corresponding first processor **12** (**12a** or **12b**). Each of the input data streams **24** (**24a**, **24b**) is formed of ingress data packets. The aggregation module **18** is adapted to analyze and combine the plurality of input data steams **24** (**24a**, **24b**) into one aggregated data stream **26** in response to priority factors of the ingress data packets. The memory **20** is adapted to stores analyzed data packets. The memory **20** may be an external buffer memory. The aggregated data stream **26** is output from the output data port **22** to the second processor **14**. Although FIG. 1 show two first processors **12**, the number of the first processors and the corresponding data streams is not limited to two.

[0014] Each of the ingress data packets includes, typically in its header, certain information such as indication of the type of the packets (ordinary data packet, protocol packet, control or management packet, and the like), port information, virtual LAN

(VLAN) address, and the like. In accordance with one embodiment of the present invention, the information indicating the data packet is a certain protocol packet is used as a priority factor. In addition, port information and VLAN information may also be used as priority factors.

[0015] In accordance with one embodiment of the present invention, each of the first processors **12** and second processors **14** includes a logical interface providing logical interconnection between a Media Access Control sublayer (MAC) and a Physical layer (PHY), such as the 10 Gigabit Media Independent Interface (XGMII), through which data streams are received and transmitted. For example, the first processors **12** may be Layer-2 switching processors implementing Ethernet Maida Access Controllers and supporting the GMII, and the second processor **14** may be a data packet processor processing the aggregated packet data stream in the GMII format. Typically, the first processors **12** receive a receive (Rx) signal as the input data stream from transceivers, and the data flow from the first processors **12** to the second processor **14** through the aggregation module **18** forms a receive data path in the system. On the other hand, the data flow form the second processor **14** to the first processors **12** typically forms a transmit (Tx) data path.

[0016] Accordingly, in accordance with one embodiment of the present invention, as shown in FIG. 1, the circuit **10** further includes an egress data input port **28** adapted to receive a data stream **30** from the second processor **14**, a forwarding module **32**, and a plurality of egress data output ports **34** (**34a**, **34b**) for outputting output data streams **36** (**36a**, **36b**) to the corresponding first processors **12**. The data stream **30** from the second processor **14** is formed of egress data packets. The forwarding module **32** is coupled between the egress data input port **28** and the egress data output ports **34**, and forwards an egress data packet in the data stream **30** to one of the egress data output port **34** in response to destination information associated with the egress data packet. The forwarding module **32** may be implemented using a field programmable logic device (FPLD), field programmable gate array (FPGA), and the like.

[0017] FIG. 2 schematically illustrates an example of implementation of the aggregation module **18** of the circuit **10** in accordance with one embodiment of the present invention. The same or corresponding elements in FIGS. 1 and 2 are denoted by the same numeral references. In this implementation, the ingress data ports **16** include a first data port **16a** for receiving a first input data stream **24a** and a second data port **16b** for receiving a second input data stream **24b**. As shown in FIG. 2, the aggregation module **18** includes a first packet analyzer **40a**, a second packet analyzer **40b**, a queue module **42**, a memory interface **44**, and an output module **46**. It should be noted that the number of the ports and the data streams is not limited to two.

[0018] The first packet analyzer **40a** is coupled to the first data port **16a**, and adapted to classify each of the ingress data packets in the first data stream **24a** into one of predetermined priority classes based on the priority factors of the ingress data packets. Similarly, the second packet analyzer **40b** is coupled to the second data port **16b**, and adapted to classify each of the ingress data packets in the second data stream **24b** into one of predetermined priority classes based on the priority factors. As described above, each of the ingress data packets includes, typically in the header, certain information such as indication of the type of the packets (ordinary data packet, protocol packet, control or management packet, and the like), port information, virtual LAN (VLAN) address, and the like, which can be used as priority factors. The priority class of each data packet is determined using one or more priority factors.

[0019] The queue module **42** includes a plurality of priority queues **48** and selection logic **50**. Each of the priority queues **48** is provided for the corresponding priority class, and the selection logic **50** implements a queue scheme. For example, four (4) priority queue may be provided. The first and second packet analyzers **40a** and **40b** analyze and classify each of the ingress data packets into one of the priority classes based on the priority factors, and also generate a packet descriptor for each of the analyzed ingress data packets. The analyzed data packet is stored in the memory **20**. The packet descriptor contains a reference to a memory location of its analyzed data packet. The packet descriptor is placed in a priority queue **48** corresponding to the priority class of the data packet. The selection logic **50** arbitrates and select a packet descriptor from among the priority queues **48** in accordance with the queue scheme. Such a queue scheme includes strict fair queuing, weighted fair queuing, and the like.

[0020] The memory interface **44** provides access to the external buffer memory **20**, and may include a first write interface **52a**, a second write interface **52b**, and a common read interface **54**. The first write interface **52a** is coupled to the first packet analyzer **40a** and adapted to write the analyzed data packets into the memory **20** at the memory location indicated by the corresponding packet descriptor. Similarly, the second write interface **52b** is coupled to the second packet analyzer **40b**, and adapted to write the analyzed data packets into the memory **20** at the memory location indicated by the corresponding packet descriptor. The common read interface **54** is coupled to the queue module **42** (the queue selection logic **50**) and adapted to read a data packet from a memory location of the memory **20** indicated by the selected packet descriptor. The data packet read from the memory **20** is provided to the output module **46** which sends the data packets to the output data port **22** as the aggregated data stream. Providing separate write interfaces (and the corresponding write ports) and a common read interface (and the corresponding common read port) saves the number of input/output (I/O) pins of the circuit **10**.

[0021] In the above-discussed embodiments, two or more input data streams from different processors are aggregated into one data stream. The present invention is also applicable when data from one processor (first processor) is oversubscribed by another (second processor), for example, when the first processor's uplink bandwidth (capacity) is greater than the second processor's data processing bandwidth (capacity). The circuit in accordance with the present invention can "bridge" the two processors and provides aggregation scheme for the oversubscribed data.

[0022] FIG. 3 schematically illustrates a circuit **11** for aggregating an input data stream from a first processor **13** into an aggregated data stream for a second processor **15**, in accordance with one embodiment of the present invention. The circuit **11**, the first processor **13**, and the second processor **15** are provided on an electronic circuit substrate.

Similarly to the circuit **10** described above, the circuit **11** includes an ingress data port **17**, an aggregation module **19**, a memory **21**, and an output data port **23**. The ingress data port receives the input data stream **25** from the first processor **13** via a first data link having a first bandwidth. Similarly to the input data stream in the circuit **10** above, the input data stream **25** is formed of ingress data packets, and each ingress data packet includes priority factors coded therein. The aggregation module **19** is coupled to the ingress data port **17**. The aggregation module **19** analyzes and selectively recombines the ingress data packets in response to the priority factors so as to generate an aggregated data stream **27** for a second data link which has a second bandwidth smaller than the first bandwidth. The memory **21** is coupled to the aggregation module **19**, and is adapted to store analyzed data packets. The output data port **23** is coupled to the aggregation module **19**, and outputs the aggregated data stream **27** to the second processor **15**.

[0023] The implementation of the circuit **11** can be done in a similar manner as that of the circuit **10** shown in FIG. 3 or circuits described in the following embodiments. One packet analyzer may be provided for the ingress data port **17**, instead of two or more packet analyzers provided for respective ingress data ports in FIG. 1 or 2, so long as the packet analyzer can handle the first bandwidth of the input data stream. Alternatively, the input data stream **25** may be divided to be handled by two or more packet analyzers. In this embodiment, the aggregation module **19** selectively recombines the stored data packet using the packet descriptors in the priority queues according to the implemented queue scheme. The above-described aggregation scheme classifying and prioritizing ingress data packets, as well as that in the following embodiments, is equally applicable to the circuit **11**. The resulting output data stream is outputted within the second bandwidth (capacity) of the second data link.

[0024] FIG. 4 schematically illustrates a circuit **100** for aggregating a plurality of input data streams from first processors into one data stream for a second processor in accordance with one embodiment of the present invention. The circuit **100**, the first processors, and the second processor are provided on an electronic circuit substrate. For example, such an electronic circuit substrate may be a circuit board for a line card, network interface device, and the like.

[0025] Similarly to the circuit **10** in FIGS. 1 and 2, the circuit **100** includes a plurality of ingress data ports **116 (116a, 116b)**, an aggregation module **118** coupled to the plurality of ingress data ports **116**, a memory **120** coupled to the aggregation module **118**, and an output data port **122** coupled to the aggregation module **118**. Each of the ingress data ports **116** receives an input data stream **124 (124a** or **124b)** from a corresponding first processor (not shown). Each of the input data streams **124 (124a, 124b)** is formed of ingress data packets, and each of the ingress data packets includes priority factors coded therein. The aggregation module **118** is adapted to analyze and combine the plurality of input data steams **124 (124a, 124b)** into one aggregated data stream **126** in response to the priority factors. The memory **120** is adapted to stores analyzed data packets. The memory **120** may be an external buffer memory. The aggregated data stream **126** is output from the output data port **122** to the second processor (not shown). Although the number of the input data streams is not limited to two, the following description uses an example where two input data streams **124** are aggregated into one data stream **126**.

[0026] As shown in FIG. 4, the ingress data ports **116** (**116a**, **116b**), the aggregation module **118**, the memory **120**, and the output data port **122** are in the receive signal (Rx) path. The circuit **110** further includes, in the transmit (Tx) data path, an egress data input port **128** for receiving a data stream **130** from the second processor (not shown), a forwarding module **132**, and egress data output ports **134** (**134a**, **134b**) for outputting output data streams **136** (**136a**, **136b**) to the corresponding first processors (not shown). The data stream **130** is formed of egress data packets. The forwarding module **132** is coupled between the egress data input port **128** and the egress data output ports **134,** and adapted to forward an egress data packet in the data stream **130** to one of the egress data output ports **134** (**134a** or **134b**) in response to destination information associated with the egress data packet. The aggregation module **118** and the forwarding module **132** may be implemented by a field programmable logic device (FPLD), field programmable gate array (FPGA), and the like.

[0027] As described above, each of the first processors and second processors may include a logical interface providing logical interconnection between a Media Access Control sublayer (MAC) and a Physical layer (PHY), such as the 10 Gigabit Media Independent Interface (XGMII), through which data streams are received and transmitted. For example, the first processors may be Layer-2 switching processors implementing Ethernet Maida Access Controllers and supporting GMII, and the second processor may be a data packet processor processing the aggregated packet data stream. Typically, the first processors receive a receive signal (Rx) as the input data stream from transceivers. For example, the first processors may be a 10GbE switching processor that supports various features used for switching and forwarding operation of data packets as well as the interface standards such as IEEE 1000BASE-T. Typically, such a 10GbE switching processor has ten or more Gigabit ports and a 10Gigabit uplink. For example, BCM 5632 processors, available from Broadcom Corporation, Irvine, California, may be used as such switching processors. However, any other MAC/PHY devices supporting required features can be used in the embodiment of the present invention. The second processor is typically a proprietary packet processor implementing specific packet processing processes and switching fabrics.

[0028] As shown in FIG. 4, the aggregation module **118** includes a first packet analyzer **140a**, a second packet analyzer **140b**, a queue module **142**, a memory interface **144** including a first memory interface **144a** and a second memory interface **144b**, and an output module **146**. The first packet analyzer **140a** is coupled to the first data port **116a,** the first memory interface **144a**, and the queue module **142**. Similarly, the second packet analyzer **140b** is coupled to the second data port **116b**, the second memory interface **144b**, and the queue module **142**. The first and second packet analyzers **140a** and **140b** analyze and classify each of the ingress data packets into one of the priority classes based on the priority factors contained in the ingress data packet. The first and second packet analyzers **140a** and **140b** also generate a packet descriptor for each of the analyzed ingress data packets. The analyzed data packets are store in the memory **120**.

[0029] As shown in FIG. 4, the external memory **120** may include a first memory unit (memory bank) **120a** and a second memory unit (memory bank) **120b** for the first input data stream **124a** and the second input data stream **124b**, respectively. In addition, the memory interface **140** may also include a first memory interface **140a** for the first input data stream **124a** and a second memory interface **140b** for the second input data stream **124b**. Each of the memory unit may include a set of quad data rate (QDR) random access memories (RAMs) as shown in FIG. 4. It should be noted that write ports for the memory units **120a** and **120b** may be provided separately for the first and second input data streams **124a** and **124b**, and a read port may be common to both the first and second input data streams **124a** and **124b**.

[0030] The packet descriptor contains a reference to a memory location of its analyzed data packet in the memory **120**. The packet descriptor is placed in the queue module **142**. The queue module **142** includes a plurality of priority queues **148** and selection logic **150**. Each of the priority queues **148** is provided for the corresponding priority class, and the packet descriptor is placed in the priority queue **148** corresponding to the priority class of its data packet. That is, packet descriptors of the ingress data packets for both of the first and second input data streams **124a** and **124b** are placed in the same priority queue **148** if they belong to the same priority class. The selection logic **150** implements a queue scheme, and arbitrates and select a packet descriptor from among the priority queues **148** in accordance with the queue scheme. Such a queue scheme includes strict fair queuing, weighted fair queuing, and the like.

[0031] The memory interface **144** provides access to the external memory **120**. When the analyzed data packets are to be written into the memory **120** (memory unit **120a** or **120b**), the first or second packet analyzer **140a** or **140b** uses the corresponding memory interface **144a** or **144b**. When the stored data packet specified by a selected packet descriptor is to be read from the referenced memory location in the memory **120**, one of the first and second interfaces is commonly used (the first interface **144a** in this example) as the read interface. The data packet read from the memory **120** is provided to the output module **146** which sends the data packets to the output data port **122** as the aggregated data stream.

[0032] As shown in FIG. 4, the first packet analyzer **140a** may include a first data decoder **150a** coupled to the first ingress data port **116a**. The first packet decoder **150a** is adapted to decode each ingress data packet to extract the priority factors therefrom. Similarly, the second packet analyzer **140b** may include a second data decoder **150b** coupled to the second ingress data port **116b**. The second packet decoder **150b** is adapted to decode each ingress data packet to extract the priority factors therefrom. For example, these packet decoders are XGMII decoders suitable to decode and extract various information (typically contained in the headers) from the ingress data packet complying the specified interface format.

[0033] As described above, the priority factors include information indicating the type of the packets (ordinary data packet, protocol packet, control or management packet, and the like), destination port information, virtual LAN (VLAN) address, and the like. In accordance with one embodiment of the present invention, the information indicating that the data packet is a certain protocol packet is used for protocol-filtering to classify certain protocols. The data packets meet the protocol filter criterion may be given the highest priority such that protocol packets are less likely to be dropped or discarded. The port information and/or VLAN information is also used as priority factors.

[0034] In accordance with one embodiment of the present invention, the priority of a data packet is assigned using per-port priority, VLAN priority, and protocol filter. For example, assume that the ingress data packets are to be classified into four priority classes. Each priority factor of an ingress data packet may be assigned with a certain number such as 3, 2, 1, or 0, indicating the priority class, with number 3 indicating the highest priority. For example, each port number may be mapped onto one of the priority numbers. If the ingress data packet has been formatted with another priority queue scheme, such an external priority number, for example, a predefined VLAN priority number, may also be mapped onto one of the (internal) priority numbers 3, 2, 1, and 0. If the ingress data packet is a protocol packet, the priority factor associated with the protocol filter may be assigned with number 3. Then, the priority numbers assigned to respective factors of the data packet are "merged" or compared each other and the highest priority number is determined as the ultimate priority number for that data packet. The data packet is classified according to the ultimate priority number. For example, if the ingress data packet is a protocol packet, it would be classified into the highest priority class even if other priority factors receives lower priority number.

[0035] Referring back to FIG. 4, the aggregation module **118** may further include a first write buffer **152a** coupled between the first data decoder **150a** and the first memory interface **144a**, and a second write buffer **152b** coupled between the second data decoder **150b** and the second memory interface **144b**. These write buffers **152a** and **152b** are typically first-in first-out (FIFO) buffers and adapted to store the analyzed data packets until they are written into the memory **120**. In accordance with one embodiment of the present invention, the aggregation module **118** may further include a flow control module **154**. The flow control module **154** monitors the first write buffer **152a** and the second write buffer **152b**, and asserts a flow control signal if an amount of data stored in the first write buffer **152a** or the second write buffer **152b** exceeds a threshold. The flow control module **154** may also monitor the priority queues **148** in the queue module **142**, and assert a flow control signal if an amount of data stored in a priority queue **148** exceeds a threshold. The flow control signal may be sent via the second processor (packet processor) to a module that controls transmit signals, and actual flow control may be done through the transmit signal path. For example, a pause control packet for the first processors is inserted in the data stream **130** such that the uplink data flow (input data streams **124**) from first processors is paused.

[0036] The output module **146** may include a read buffer **156** coupled to a common read interface of the memory interface **144**, and a data encoder **158** coupled to the read buffer **146**. The data encoder **158** encodes the data packets into an interface format corresponding to that used by the first and second processors. For example, the data packets are encoded into the XGMII format to form an output data stream sent from the output data port **122**.

[0037] As shown in FIG. 4, in the transmit signal (Tx) path, the circuit **110** includes the forwarding module **132** between the egress data input port **128** and the egress data output ports **134a** and **134b**. In accordance with one embodiment of the present invention, the forwarding module **132** includes a data decoder **160**, a buffer **162**, first and second forwarding logic **164a** and **164b**, and first and second data encoders **166a** and **166b**. The forwarding logic **164a** and **146b** forwards an egress data packet of the data stream **130** to one of the data encoders **166a** or **166b** in response to destination information associated with the egress data packet.

[0038] FIG. 5 schematically illustrates an example of a system **200** in which two data streams from the switching processors **202** are aggregated into one data stream for a packet processing processor (XPP) **204** by an aggregation circuit **206** in accordance with one embodiment of the present invention. For example, the system **200** may be 60 Gigabit over Copper (60 GoC) line card, and the switching processors **202** may be Broadcom's BCM5632s explained above. The aggregation circuit **206** may be one of the circuits **10**, **11**, or **110** as described in embodiments above. As shown in FIG. 5, the system **200** includes thee sets (stacks) of aggregation data pipe lines **208** (**208a**, **208b**, and **208c**). In each of the data pipe lines **208**, the aggregation circuit **206** bridges two of the switching processors **202** to one packet processing processor **204**. The data coupling between the switching processors **202** and the aggregation circuit **206**, and that between the aggregation circuit **206** and the packet processor **206** are supported by the XGMII. Each of the switching processors **202** receives ten (10) Gigabit data streams from Gigabit Ethernet transceivers **210**, for example, BCM5464 Quad-Port Gigabit Copper Transceivers, available from Broadcom Corporation, Irvine, California. The data aggregation of the oversubscribed input data is performed in the lower layers (PHY/MAC), prior to actual packet processing in higher layers.

[0039] FIG. 6 schematically illustrates a method for aggregating a plurality of input data streams from first processors into one data stream for a second processor in accordance with one embodiment of the present invention. The first processors and the second processor are provided on an electronic circuit substrate. The method may be performed by the circuits **10**, **11**, **110**, or **204** described above.

[0040] An input data stream is received from each of the first processors (**300**). Each input data stream is formed of ingress data packets, and each ingress data packet includes priority factors coded therein, as described above. Each of the ingress data packets are analyzed and classified into one of predetermined priority classes based on the priority factors (**302**). The analyzed ingress data packet is stored in a memory (**304**), and a packet descriptor is generated for the analyzed ingress data packet (**306**). The packet descriptor contains a reference to a memory location of its analyzed data packet stored in the memory. The packet descriptor is placed in a priority queue corresponding to the priority class of the data packet (**308**). The packet descriptors from each data stream of the same priority class are placed in the same priority queue for that priority class. A packet descriptor is selected from among the priority queues by arbitrating the packet descriptors in the priority queues using selection logic implementing a queue scheme (**310**). A data packet corresponding to the selected packet descriptor is read from the memory (**312**), and an aggregated data stream is generated combining the data packets read from the memory, and aggregated data stream is sent to the second processor (**314**).

[0041] FIG. 7 schematically illustrates the method of aggregating a plurality of data streams along the receive (Rx) data path in accordance with one embodiment of the present invention. The input data streams (two data streams in this example) from switching processors (first processors) are received at the respective receive signal (Rx) front ends (**320a** and **320b**), and a header of each ingress data packet is decoded to extract the priority factors. The data format may be that of the XGMII. Ingress data packets are buffered in the corresponding write buffers (**322a** and **322b**) during the packet analysis until they are stored in the memory. The write buffers may be QDR FIFOs. The ingress data packets are evaluated and classified into different priority classes in accordance with the priority factors (**324a** and **324b**). The packet descriptors and analyzed ingress data packets are sent to the write interfaces (**326a** and **326b**). The packet descriptors are placed into the priority queues **328** corresponding to the priority class of its ingress data packet. For example, four (4) priority queues are provided. The analyzed ingress data packets are stored in the corresponding buffer memories (**330a** and **330b**). The buffer memories may be external QDR RAMs. The packet descriptors in the priority queues are arbitrated by queue selection logic (**332**), and the selected packet descriptor is sent to the read interface (**334**). Since the packet descriptor includes a reference to the memory location of its data packet, the corresponding data packet is read from the memory through the read interface. The read-out data packets are buffered in a read FIFO (**336**), and then encoded into the specific data format (**338**), for example that of the XGMII. The encoded data packets are sent as an output data stream to the second processor (packet processor)

[0042] As shown in FIG. 7, write-buffering, analyzing and classifying, and storing the data packets, and generating packet descriptors are performed separately for each data stream (**320a** through **326a**, and **330a**; **320b** through **326b**, and **330b**). However, the packet descriptors for the both data streams are stored in the common priority queues and commonly arbitrated (**328**, **332**). The stored data packet specified by the selected packet descriptors are also read out using the common read interface, and the data packets thereafter are processed in a single data channel (**334** through **338**). As described above, in analyzing and evaluating the ingress data packets, protocol-filtering, per-port priority, VLAN priority, and the like may be used as priority factors.

[0043] FIG. 8 schematically illustrates the method of aggregating a plurality of data streams along the transmit (Tx) data path in accordance with one embodiment of the present invention. A data stream formed of egress data packets from a packet processor (second processor) is received at a transmit signal (Tx) front end (**340**) and decoded to extract their destination information. The decoding may include decoding a specific interface data format such as the XGMII into a single data rate (SDR). The decoded data packets are buffered in a FIFO (**342**), and dispatched to the destination port by forwarding logic (**344**). Since one data stream is divided into two output data streams for different switching processors, an Idle Packet is inserted between End of Packet (EOP) and Start of Packet (SOP) in each data stream, such that the data for the other destination is replaced with the idle data (**346a** and **346b**). Each of the output data stream is encoded for an interface format such as the XGMII (**348a** and **348b**).

[0044] FIG. 9 schematically illustrates a method for aggregating a plurality of input data streams from first processors into one data stream for a second processor, in accordance with one embodiment of the present invention. The first processors and the second processor are provided on an electronic circuit substrate. A field programmable logic device (FPLD) coupled between the first processors and the second processor is provided (**350**). An ingress data interface is provided between each of the first processors and the FPLD (**352**). Each ingress data interface is adapted to couple an input data stream from a corresponding first processor to the FPLD. For example, the ingress data interface may be the XGMII supported by the first processor. Each input data stream is formed of ingress data packets, and each ingress data packet includes priority factors coded therein, as described above. An output data interface is also provided between the FPLD and the second processor (**354**), which is adapted to couple the aggregated data stream to the second processor. For example, the output data interface may be a XGMII supported by the second processor. A memory coupled to the FPLD is also provided (**356**), which is adapted to store analyzed data packets. The FPLD is programmed such that the FPLD analyzes and combines the plurality of input data steams into one aggregated data stream in response to the priority factors (**360**). The programmed FPLD performs the aggregation function for the Rx data steam as described above in detail with respect to other embodiments. The FPLD may also programmed such that it also performs forwarding functions for the Tx data stream as described above, with providing an input data interface for receiving the Tx data from the second processor, and output interfaces for outputting output data streams to the first processors.

[0045] The numbers of ports, processors, priority queues, memory banks, and the like are by way of example and are not intended to be exhaustive or limiting in any way. While embodiments and applications of this invention have been shown and described, it would be apparent to those skilled in the art having the benefit of this disclosure that many more modifications than mentioned above are possible without departing from the inventive concepts herein. The invention, therefore, is not to be restricted except in the spirit of the appended claims.